



Article

High-Capacity Data Hiding for ABTC-EQ Based Compressed Image

Cheonshik Kim ^{1,*},† , Ching-Nung Yang ^{2,†} and Lu Leng ^{3,4,*} ¹ Department of Computer Engineering, Sejong University, Seoul 05006, Korea² Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien 97401, Taiwan; cnyang@gms.ndhu.edu.tw³ Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang 330063, China⁴ School of Electrical and Electronic Engineering, College of Engineering, Yonsei University, Seoul 120749, Korea

* Correspondence: mipsan@sejong.ac.kr (C.K.); drluleng@gmail.com (L.L.)

† These authors contributed equally to this work.

Received: 27 February 2020; Accepted: 8 April 2020; Published: 14 April 2020



Abstract: We present a new data hiding method based on Adaptive BTC Edge Quantization (ABTC-EQ) using an optimal pixel adjustment process (OPAP) to optimize two quantization levels. The reason we choose ABTC-EQ as a cover media is that it is superior to AMBTC in maintaining a high-quality image after encoding is executed. ABTC-EQ is represented by a form of $trio(Q_1, Q_2, [Q_3], BM)$ where Q is quantization levels ($Q_1 \leq Q_2 \leq Q_3$), and BM is a bitmap. The number of quantization levels are two or three, depending on whether the cover image has an edge or not. Before embedding secret bits in every block, we categorize every block into smooth block or complex block by a threshold. In case a block size is 4×4 , the sixteen secret bits are replaced by a bitmap of the smooth block for embedding a message directly. On the other hand, OPAP method conceals 1 bit into LSB and 2LSB respectively, and maintains the quality of an image as a way of minimizing the errors which occur in the embedding procedure. The sufficient experimental results demonstrate that the performance of our proposed scheme is satisfactory in terms of the embedding capacity and quality of an image.

Keywords: data hiding; ABTC-EQ; AMBTC; BTC; LSB; OPAP

1. Introduction

The Internet is deeply involved in everyday life and work; most work is actively introducing Internet-based cloud services in modern society. This means that personal and corporate data will no longer be stored only on personal computers. Accordingly, security issues for digital data are becoming more important. The watermark may be copyright information and authentication information to determine the copyright owner of the work, to authenticate the authenticity and integrity of the multimedia work, and to achieve the purpose of copyright protection. Digital watermarking technology has several features such as digital media protection, authentication, distribution, and anti-counterfeiting. Therefore, the research in this area is important [1].

While data hiding (DH) [2] is a method to transmit information to the destination secretly by embedding information in cover media. Unlike existing communication methods, a hidden exchange is secure communication. Thus, DH has a potential for safety communication. DH does not replace encryption, but it may be a safety personal communication when it cannot be detect by attackers. If the cover image embeds secret data, the cover image is distorted and cannot be recovered. In some

applications, such as military imaging systems and remote sensing, it is desirable to restore the stego media back to the original media without distortion. The DH that has this feature of reversibility is called reversible DH (RDH) [3].

DH may be classified into two technologies in the spatial domain and the frequency (or transform) domain. In the spatial domain, the secret message is embedded into the pixels of the cover image directly. DH based on the Least Significant Bit (LSB) [4–6] is mainly used in the spatial domain. In the frequency domain, the cover image must first be transformed using Discrete Cosine Transform (DCT) [7,8] Discrete Wavelet Transform (DWT) or similar frequency-oriented mechanisms, and then secret message is embedded in such a transformed domain [9]. Vector Quantization (VQ) [10] and block truncation coding (BTC) [11] are lossy compression methods, and many DH methods based on the compressions have been proposed.

After BTC, lossy compression methods such as Absolute Moment Block Truncation Coding (AMBTC) [12–14], Modified BTC (MBTC) [15], and Adaptive BTC based on Edge Quantization (ABTC-EQ) [16] with improved performance appeared. These compression methods also have the advantages of a stable compression ratio and low computation time, which are suitable for wide use in DH and watermarking. The compressed basic structure of AMBTC is $trio(Q_1, Q_2, BM)$, where Q_1 and Q_2 are two quantization levels, and BM is a bitmap. Here, a is less than b . One of the most efficient methods in AMBTC-based DH is to embed secret data directly into the bit-plane. Chuang and Chang [17] introduced a way to embed data using Direct Bitmap Substitution (DBS). Here, a suitable condition to embed secret data is to use smooth blocks rather than complex blocks because it helps to maintain good image quality.

Chen et al. [18] introduced a method of the Order of Two Quantization Levels (OTQL) to embed data into $trio$ compressed using AMBTC without loss. They may hinder the loss of the stego block by using OTQL and flipped bitmap BM. That is why the OTQL method is efficient. Ou and Sun [19] pointed out that flipping the bitmap BM for hiding secret data without adjusting the coefficients of quantization could result in distortion. Besides, to improve the quality of the stego image, they introduced an adjusting method for the quantization levels.

Huang et al. [20] introduced a hybrid method increasing as much as \log_2^T bits by using difference expansion and the reduction of two quantization levels. The merit of this method is that it might not need the type of the blocks. Besides, they also introduced OTQL for higher Embedding Capacity (EC) without loss additionally. Hong [21] introduced a method to embed data in the quantization level of smoothing blocks using Adaptive Pixel Pair Matching (APPM) techniques [22]. Chen and Chi [23] processed by sub-dividing less complex block and highly complex block. In 2016, Malik et al. [24] introduced a DH based on AMBTC using 2-bit plane and 4 quantization levels. The merit of this method is high payload, and the demerit is to decrease the compression rates.

In this paper, we propose a new DH method based on ABTC-EQ with high embedding rates and good image quality. Because in the existing DH based on AMBTC the quality of the cover image is not high compared to the original image, there is a problem that the PSNR of the cover image decreases below 30 dB when we embed many bits into the cover image, and we replace AMBTC with ABTC-EQ in order to solve this problem. Our purpose is to increase the performance of DH by applying the LSB method to the pixels of quantization levels. There exists a demerit to generate some noises in encoding procedure, though the LSB substitution method is enough to achieve our purpose. In order to solve the problem, we propose a way to optimize the errors and enhance the quality of an image by using Optimal Pixel Adjustment Process (OPAP) [6] in this paper.

The rest of this paper is organized as follows. Section 2 briefly reviews AMBTC, ABTC-EQ, LSB substitution and OPAP, and Ou & Sun's method. The proposed new DH based on ABTC-EQ is introduced in Section 3. Experimental results and discussions are given in Section 4. And Section 5 sketches some conclusions.

2. Related Works

2.1. AMBTC

Lema and Mitchell introduced AMBTC, a variant of BTC which improved BTC's computation time. Moreover, the quality of the AMBTC is better than that of the BTC. AMBTC obtains the high and low mean of each block and uses these two values to produce a compressed size of grayscale image. In AMBTC, the image is first divided into non-overlapped blocks sized $m \times m$, where $m \times m$ may be set to (4×4) , (8×8) , (16×16) , and so on. AMBTC works on block by block, and the mean value for each block is calculated using Equation (1).

$$\bar{x} = \frac{1}{m \times m} \sum_{i=1}^{m^2} x_i, \quad (1)$$

Here, x_i represents the i^{th} pixel value in this block. For every pixel in each block, binary values ('0' or '1') are produced using Equation (2), and a block composed of '0's or '1's is called a bitmap block. That is, a value larger than the value \bar{x} is replaced by '1', otherwise replaced by '0'.

$$b_i = \begin{cases} 1, & \text{if } x_i \geq \bar{x}, \\ 0, & \text{if } x_i < \bar{x}, \end{cases} \quad (2)$$

Let t denote the number of '1' in the bitmap BM. The mean Q_2 or Q_1 above or below of \bar{x} is calculated in Equation (3).

$$Q_2 = \left\lfloor \frac{1}{t} \sum_{x_i \geq \bar{x}} x_i \right\rfloor \text{ and } Q_1 = \left\lfloor \frac{1}{(m \times m) - t} \sum_{x_i < \bar{x}} x_i \right\rfloor. \quad (3)$$

The two quantization levels and the bitmap BM can be represented by the basic unit *trio* ($Q_1 \neq 0, Q_2 \neq 1, \text{BM}$) of compression. The compressed *trio* is used for decoding to reconstruct the grayscale image. For $m = 4, 16$ pixels are represented by a *trio* (Q_1, Q_2, BM) of $8 + 8 + 16 = 32$ bits, so the compression ratio (CR) is $(16 \times 8) / 32 = 4$. For 512×512 pixel images, the file size of 2 M-bits can be reduced to 0.5 M-bits. AMBTC provides better image quality and faster computation, so most BTC-based DH and secret image sharing schemes adopt the AMBTC. Mathews et al. [15] proposed MBTC using a maximum and minimum quantizer to further improve the quality of reconstructed images. AMBTC divides the pixels in a block into two groups ('0's or '1's) using the mean of the block, and based on this, obtains two quantization levels. In Equation (4), the threshold value x^{th} of MBTC is obtained by the mean of the maximum (x_{\max}), the minimum (x_{\min}) and the mean (\bar{x}) in a block.

$$x^{\text{th}} = \frac{(x_{\max} + x_{\min} + \bar{x})}{3}. \quad (4)$$

In MBTC, a *trio* is obtained by using x^{th} instead of the mean of per block. It is a difference between AMBTC and MBTC.

2.2. ABTC-EQ

AMBTC and MBTC in BTC-like obtain two quantization levels in per block using mean values according to the given equations. ABTC-EQ, on the other hand, uses a canny edge detector [25] to find edge from the input image and to obtain quantization levels based on it. To extract two quantization levels, an edge information, $E = [e_i]$, for each block ($m \times m$ size) is extracted, which is classified into edge block or non-edge blocks. If E in the block is all zeros, it is a non-edge block, otherwise it is defined as an edge block. We use two quantization level approaches to non-edged block. For non-edge blocks, each block can be represented as a *trio* (Q_1, Q_2, BM_n) using the MBTC method, where BM_n notation is

used intentionally to represent the bitmap for non-edge blocks. For edge blocks, three cluster (c_0, c_1, c_2) are found and grouped using the k -means cluster algorithm [26]. The bit map $BM_e = [b_i]$ of the edge block is generated by Equation (5).

$$b_i = \begin{cases} 00, & \text{if } x_i \in c_0, \\ 01, & \text{if } x_i \in c_1, \\ 10, & \text{if } x_i \in c_2. \end{cases} \quad (5)$$

The mean $Q_i (1 \leq i \leq 3)$ of each cluster c_i is calculated by Equation (6), where $Q'_1 \leq Q'_2 \leq Q'_3$. Thus, the edge image block is represented by (Q'_1, Q'_2, Q'_3, BM_e) .

$$\mu'_i = \left\lfloor \frac{1}{|c_i|} \sum_{x_i \in c_i} x_i \right\rfloor, 0 \leq i \leq 2. \quad (6)$$

Bi-clustering and tri-clustering methods are applied to non-edge blocks and edge-blocks. In *trio*, the identifier f is used to distinguish non-edge blocks and edge-blocks, and is defined as '1' and '0', respectively. Finally, k^2 pixels can be represented as $(f = 1, Q_1, Q_2, BM_n)$ or $(f = 0, Q_1, Q_2, Q_3, BM_n)$. Therefore, the CR in ABTC-EQ is dynamic, not static, as in the previous BTC scheme.

2.3. LSB Substitution and OPAP

Least-Significant-Bit (LSB) substitution is a way to manipulate the LSB planes by directly replacing the LSBs of the cover-image with the hidden bits. Wang et al. [27] introduced a DH by optimal LSB substitution and genetic algorithm, where the worst mean-square-error (WMSE) is shown to be 1/2 of that obtained by the simple LSB substitution technique. Let us look at the DH procedure for the original 8-bit grayscale represented by $x_i \in \{0, 1, \dots, 255\}$. S denotes n -bit hidden values represented as $S = \{s_k | 0 \leq k < n, s_k \in \{0, 1\}\}$. The mapping between the n -bit secret bits $S = \{s_k\}$ and the embedded bits $S' = s'_k$ can be defined as follows: $s'_k = \sum_{j=0}^{\delta-1} s_{k \times \delta + j} \times 2^{\delta-1-j}$. The pixel value x_i for embedding the δ -bit s'_k is changed to form the stego-pixel x'_i like $x'_i = x_i - (x_i \bmod 2^\delta) + s'_k$. The δ LSBs of the pixels are extracted by $s_k = x'_i \bmod 2^\delta$.

An Optimal Pixel Adjustment Process (OPAP) has an ability to enhance the image quality of the stego-image generated by the simple LSB substitution method and it was proved by mathematically proof. Let x_i be the pixel of the cover image, x'_i be the obtained pixel from pixel x_i using the LSB replacement method, and x''_i is the optimized pixel derived from x'_i by the OPAP method. Let $e = x'_i - x_i$ be the error between x_i and x'_i . x'_i is obtained by the direct substitution of the δ LSBs of x_i with δ secret bits. The value of Δ_i may be segmented into three intervals. The OPAP modifies x' to form the stego-pixel x'' as the following rule:

1. Rule 1 ($2^{\delta-1} < \Delta_i < 2^\delta$): if $x'_i \geq 2^\delta$, then $x''_i = x'_i - 2^\delta$; otherwise $x''_i = x'_i$;
2. Rule 2 ($-2^{\delta-1} \leq \Delta_i \leq 2^{\delta-1}$): $x''_i = x'_i$;
3. Rule 3 ($-2^\delta < \Delta_i < -2^{\delta-1}$): if $x'_i < 256 - 2^\delta$, then $x''_i = x'_i + 2^\delta$; otherwise $x''_i = x'_i$;

2.4. Ou and Sun's Method

In this method, smooth blocks of AMBTC are mainly used with DBS method in order to improve DH performance and less image distortion. In addition, the two quantization levels of the block are re-calculated in order to recover the distorted pixels after embedding secret message. For complex blocks, they embed 1-bit into *trios* by using the OTQL. The detailed procedure of embedding secret bits is described as follows:

Input: A bit sequence S , *trios*, and a parameter T .

Output: A stego AMBTC-compressed codes *trios*.

Step 1: One block (including two quantization levels and a bit-plane B_i) from the *trios* is got.

Step 2: Compute the absolute difference value d_i of two quantization levels Q_{1i} and Q_{2i} , such that $d_i = |Q_{1i} - Q_{2i}|$.

- Step 3:** If $d_i > T$, 1-bit from S can be embedded. If s_i is equal to '1', the bit plane BM_i is toggled and the two quantization levels are exchanged, $trio(Q_{2_i}, Q_{1_i}, B\bar{M}_i)$. Then, add $(Q_{2_i}, Q_{1_i}, B\bar{M}_i)$ into $trios$. If s_i is equal to '0', there is no change of the $trio$. Return to Step 1 to process the next block.
- Step 4:** If $d_i \leq T$, the 16-bits from S can be embedded. The bit-plane BM_i are substituted with the 16-bits. Re-calculate two new quantization levels Q'_{1_i} and Q'_{2_i} . Then, add $(Q'_{1_i}, Q'_{2_i}, BM_i)$ into $trios$. Return to Step 1 to process the next block.
- Step 5:** Repeat Steps 1~4 until all the blocks are completely processed. The output compressed $trios$, each of which contains two quantization levels and 1-bit plane.

3. The Proposed Scheme

This section discusses new DH method based on ABTC-EQ by using OPAP. In our proposed method, OPAP combine existing DBS and OTQL to increase the performance. Also, to improve the quality of the image, ABTC-EQ is used as the cover image. Thus, the method can embed an appropriate amount of secret data while maintaining the displayed image quality. A complete description of the proposed algorithm along with an example is given in Section 3.3.

3.1. Embedding Procedure

First, $trios$ are generated by compressing the original gray image with ABTC-EQ before applying the proposed embedding algorithm. The generated $trios$ consist of quantization levels and bitmaps. The quantization levels representing each block are Q_1, Q_2 , and Q_3 . If the image is an edged block, Q_3 is required. A bitmap $BM = \{x_i | 0 \leq i < n, x_i \in \{\{0, 1\} \vee \{0, 1, 2\}\}\}$ is divided into two groups: $\{0, 1\}$ and $\{0, 1, 2\}$. If the image is an edged block, three elements of $\{0, 1, 2\}$ are needed. The procedure of DH in our proposed method is shown in Figure 1. Three methods (OPAP, DBS, and OTQL) are applied to the generated $trios$ in order to conceal the data. The proposed method is applied to the $trios$ block repeatedly.

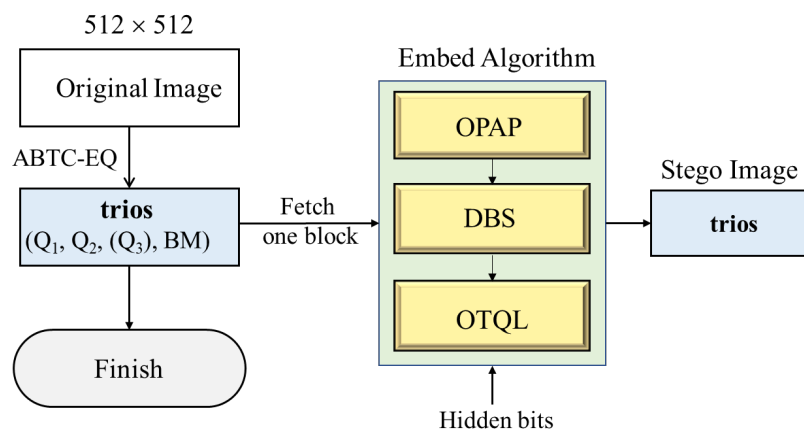


Figure 1. Schematic of the embedding procedure.

To improve DH performance, the OPAP algorithm is used for DH regardless of block type (i.e., smooth or complex). By increasing or decreasing the pixel value by one according to the pattern of the LSB, it is possible to reduce the distortion of the image while increasing the number of data concealment bits. When we use the previous DBS method to the bitmaps, only the $\{0, 1\}$ bitmap is used, and $\{2\}$ is not used for DH to reduce image distortion. The DBS method distinguishes two group, that is, smooth and complex blocks, using a threshold and the DBS method may be used only for smooth blocks. In addition, to complex blocks, we use OTQL method to hide 1-bit per block additionally. Our proposed method should apply OPAP, DBS, and OTQL in order, and keep this order

in the data extraction process. The complete proposed embedding algorithm is discussed as follows:

Input: original image I of size $M \times N$ pixels, T : threshold, S : secret bits

Output: ABTC-EQ compressed marked codes, *trios*

Step 1: Divide the original image into non-overlapping 4×4 sized blocks. Next, apply the ABTC-EQ method to compress every block. After all, we obtain two or three quantization values and one bitmap for each block, which is represented as $trio(Q_1, Q_2, [Q_3], BM)$.

Step 2: Fetch one *trio* from compressed packet (*trios*) according to the defined order. Embed 4-bits into two quantization level values, Q_1 and Q_2 , using OPAP rules (Tables 1 and 2), which is LSB optimization we proposed. Transform the value Q_1 into binary value and then embed one bit into second LSB by using the rule (Table 1 and Equation (7)) (notes, the variables S and OP are secret bit and operation, respectively. For example, given the 2LSB and LSB are '00' and S is '1', it may embed $S = '1'$ into the pixel using the calculation of the OP , ' $Q_1 - 1$ '). Then, it is possible to hide 1-bit as XOR operation between 2LSB and LSB of the pixel Q_1 and the given operation, according to the rule of Table 2 and Equation (8). For the pixel Q_2 , the same procedure as shown in the pixel Q_1 is used and it may embed 2-bits additionally.

$$Q'_1 = \begin{cases} Q_1 - 1, & \text{if } (2LSB(0) \text{ and } LSB(0)) \text{ and } S(1), \\ Q_1 + 1, & \text{if } (2LSB(0) \text{ and } LSB(1)) \text{ and } S(1), \\ Q_1 - 1, & \text{if } (2LSB(1) \text{ and } LSB(0)) \text{ and } S(0), \\ Q_1 + 1, & \text{if } (2LSB(1) \text{ and } LSB(1)) \text{ and } S(0), \\ \text{no change,} & \text{otherwise} \end{cases} \quad (7)$$

$$Q''_2 = \begin{cases} Q'_2 - 1, & \text{if } (2LSB(1) \text{ and } LSB(1)) \text{ and } S(1), \\ Q'_2 + 1, & \text{if } (2LSB(1) \text{ and } LSB(0)) \text{ and } S(0), \\ Q'_2 - 1, & \text{if } (2LSB(0) \text{ and } LSB(1)) \text{ and } S(0), \\ Q'_2 + 1, & \text{if } (2LSB(0) \text{ and } LSB(0)) \text{ and } S(1), \\ \text{no change,} & \text{otherwise} \end{cases} \quad (8)$$

Step 3: If $(|Q_{1_i} - Q_{2_i}|) \leq T$, take bitmap BM from the *trio* and apply Equation (9) to it (notes: count 0s and 1s in the BM , then take the same number of secret bits from the S). Replace every binary element of the BM with secret bits S in the specified order directly; otherwise, go to Step 2.

$$BM'_i = \begin{cases} S_i^{count(BM_i=0 \text{ or } 1)}, & \text{if } (|Q_{1_i} - Q_{2_i}| \leq T), \\ \text{no action,} & \text{otherwise.} \end{cases} \quad (9)$$

Step 4: If $(|Q_{1_i} - Q_{2_i}|) \neq 0$ and $S_i = 1$, change order of two quantization level Q_1 and Q_2 , that is, $trio(Q_2, Q_1, BM)$. If $(|Q_{1_i} - Q_{2_i}|) \neq 0$ and $S_i = 0$, it is remained unchanged.

Step 5: Steps 2 ~ 4 are repeated until all *trios* are scanned.

Step 6: Save the *trios* and return to the *main* function.

Table 1. Rule of optimal pixel adjustment process (OPAP) for second Least Significant Bit (LSB).

Pixel		S	OP	Pixel'	
Second LSB	First LSB			Second LSB	First LSB
0	0	1	$Q - 1$	1	1
0	1	1	$Q + 1$	1	0
1	0	0	$Q - 1$	0	1
1	1	0	$Q + 1$	0	0

Table 2. Rule of OPAP for first and second LSB.

Pixel'		XOR	S	OP	Pixel''	
Second LSB	First LSB				Second LSB	First LSB
1	1	0	1	$Q' - 1$	1	0
1	0	1	0	$Q' + 1$	1	1
0	1	1	0	$Q' - 1$	0	0
0	0	0	1	$Q' + 1$	0	1

3.2. Extraction Procedure

The procedure to extract data from marked *trios* is shown in Figure 2. First, read a block (*trio*) from *trios*. Then, OPAP, DBS, and OTQL are applied to the quantization level and the bitmap in order to extract the secret bits hidden in the *trios*. At this time, *trios* are divided into smooth blocks and complex blocks by using a predefined threshold T . OPAP are applied to *trios* regardless of the type (i.e., smooth or complex) of them. DBS is used on only smooth blocks. Lastly, the order of two quantization levels are used to hide 1-bit per block additionally.

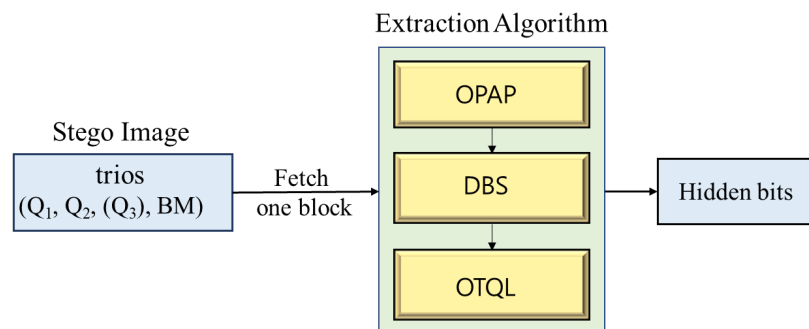


Figure 2. Schematic of the extraction procedure.

The complete proposed extraction algorithm is discussed as follows:

Input: AMBTC compressed marked codes, T is threshold

Output: S is secret data bit stream

- Step 1:** The i^{th} block is fetched from compressed packet, *trios*, according to the defined order.
- Step 2:** For the quantization values Q_1 and Q_2 of the i^{th} *trio*, Equations (10) and (11) are used for obtaining 4-bits of hidden bits. (note: the l is an indexing variable).

$$[s_{l+1}] = \left\lfloor \frac{Q_i}{2} \right\rfloor \bmod 2, \tag{10}$$

$$[s_{l+2}] = xor \left(\left\lfloor \frac{Q_i}{2} \right\rfloor \bmod 2, Q \bmod 2 \right) \tag{11}$$

- Step 3:** If $(|Q_{1_i} - Q_{2_i}|) \leq T$, Equation (12) is applied to the BM for obtaining the hidden bits. That is, the hidden bits are appended in s_l . Otherwise, go to Step 1.

$$s_l = \begin{cases} BM_i, & \text{if } (|Q_{1_i} - Q_{2_i}| \leq T \text{ and } BM_i = 0 \text{ or } 1), \\ \text{go to [Step 1]}, & \text{otherwise.} \end{cases} \tag{12}$$

- Step 4:** If $(|Q_{1_i} - Q_{2_i}|) \neq 0$ and the order of two quantization level is ascending order, then assign '0' into s_l . if $(|Q_{1_i} - Q_{2_i}|) \neq 0$ and the order is descending order, then assign '1' into s_l .
- Step 5:** Steps 1~4 are repeated until all *trios* are scanned.
- Step 6:** Return to *main* function.

3.3. Example of Embedding Process

In Figure 3a–d, the example of the proposed embedding algorithm is discussed. A secret bit stream $S = (11001100100101101010)_2$ and threshold $T = 6$ are considered to the example. Suppose there is a block $B = (101\ 100\ 96\ 94\ 101\ 100\ 96\ 94\ 101\ 100\ 96\ 94\ 101\ 100\ 96\ 94)$ of the original image. Canny edge detector confirmed B as non-edged block. Thus, we apply the procedure of Figure 3a to B . That is, first, $x^{th} = ((101 + 94 + 97))/3$ is calculated using Equation (4). If x_i is greater than or equal to x^{th} , '1' is assigned to the position of the corresponding pixel. Otherwise, '0' is assigned to the position.

Finally, the bitmap is constructed by this procedure. Two quantization levels representing the original block are obtained by using Equation (3). That is, $Q_1 = 95$ and $Q_2 = 100$. The compression type of a block represents as $trio(f, Q_1, Q_2, BM) = (1, 95, 100, 1100110011001100)$. Figure 3b shows the detailed procedure that secret bits $S = '1100_{(2)}$ ' is embedded into the two quantization levels Q_1 and Q_2 by OPAP method. Then, first, convert Q_2 and Q_1 to binary, ($Q_2 : 01100100, Q_1 : 01011111$).

We may exploit Rule 1 (Table 1, Equation (7)) to embed the bit $S_1 = '1'$ into the 2LSB of Q_1 . In case of Q_1 , there is no need to apply Rule 1 of Equation (7) since 2LSB = '1'. To embed consecutive 1-bit, we apply Equation (8) into the quantization level Q_1 . That is, we obtain $Q_1''(94)$ by the equation of $Q_1' - 1$, since (XOR (2LSB = 1, LSB = 1) = 0) and $s_1 = '1'$. For Q_2 , we also may obtain $Q_2''(100)$ by the same procedure as Q_1 . The procedure of Q_2'' concealing '00' is unnecessary, because Q_2'' already had '00'.

Lastly, we finished embedding 4-bits in two quantization levels, Q_1 and Q_2 , by using OPAP. We can see the procedure where the bitmap is replaced by the secret bits of 4×4 sized block directly by DBS method in Figure 3c. For embedding secret bits S , the bit stream of S replace the BM of the smooth block directly. Thus, we have to find the smooth blocks according to predefined threshold value T of criteria ($|Q_1 - Q_2| \leq T = |94 - 100| \leq 6$) and then apply DBS method to the blocks. In this example, this block is the smooth block because it satisfies with the given criteria.

That is, the block BM is filled with $S = (1100100101101010)$ (Figure 3c). Figure 3d may use existing OTQL method to embed 1-bit additionally. This OTQL method is recommended by having an hint from the fact that two or three quantization levels of the blocks are specific ordered when each block is compressed. In other words, if the hidden bit is $S = '1'$, then we change the order of two quantization levels. In Figure 3d, switch the order of Q_1 and Q_2 , that is, $trio(Q_2, Q_1, BM) = (1, 100, 94, 1100100101101010)$.

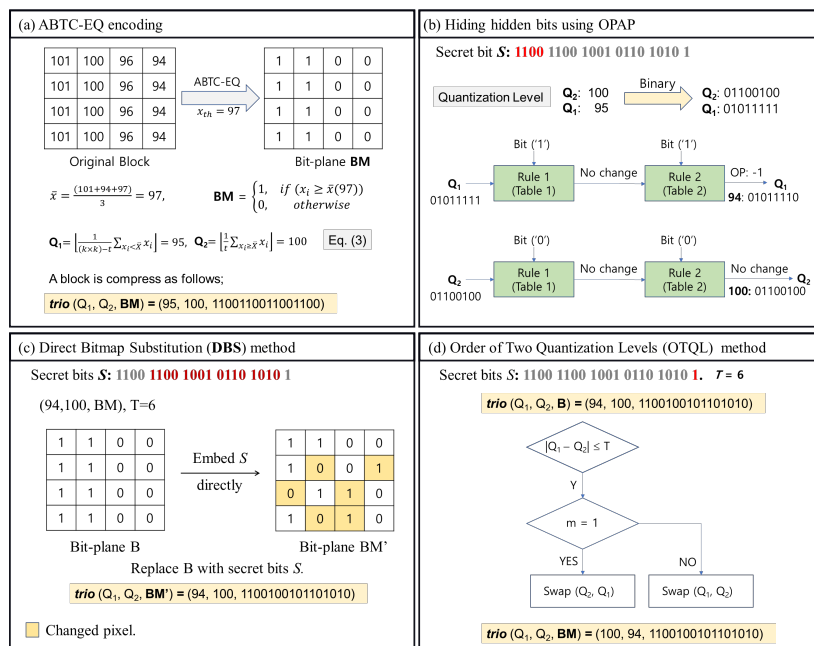


Figure 3. Example of data hiding based ABTC -Equation.

3.4. Example of Extraction Process

In this section, we introduce extraction procedure of hidden bits from the given $\text{trio}(Q_2, Q_1, BM) = (100, 94, 1100100101101010)$. First, the hidden bit '11₍₂₎' may be extracted from the $Q_1 = 94$ using Equations (10) and (11), that is, $2\text{LSB}(Q_1)$ and $2\text{LSB}(Q_1) \oplus \text{LSB}(Q_1)$, respectively. After, we might obtain the hidden values '00₍₂₎' from $Q_2 = 100$ applying Equations (10) and (11) to Q_2 , respectively. Before applying DBS method, we confirm whether or not the block is edged block using canny edge detector and k -mean. Thus, it is possible to achieve as assigning all pixels of the BM into the variable S' when $(|Q_1 - Q_2| \leq T \text{ and } BM = 0 \text{ or } 1)$. Finally, we get the hidden bits S with this decoding procedure, that is, $S' = (11001100100101101010)_2$

4. Experimental Results and Discussions

In this section, we compare and analyze the experimental results of the proposed and other existing methods. For a comparison and analysis, we use nine grayscale images of size 512×512 pixels, as shown in Figure 4. The secret data used in the experiments are random bits generated by the Pseudo Random Number Generator (PRNG). For evaluation, EC and peak signal-to-noise ratio (PSNR) are used to compare the performance of the existing and proposed methods. The DH capacity represents the number of secret bits contained in the cover image. The quality of the image is measured by the PSNR defined as:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}. \quad (13)$$

In other words, PSNR is the ratio between the maximum possible power of a signal and the power of noise that affects the fidelity of its representation. The MSE used for the PSNR is the average intensity difference between the compressed and the reference image. If the MSE value of the compressed image is low, the image quality is evaluated as good. MSE is calculated used the reference image p and the distorted image p' as follows.

$$\text{MSE}(p, p') = \frac{1}{N} \sum_{i=1}^N (p_i - p'_i)^2. \quad (14)$$

The error value $\varepsilon = p_i - p'_i$, denotes the difference between the original and the distorted signal. The 255^2 means the allowable pixel intensity in Equation (13).

Structural Similarity Index Metric (SSIM) [22] is used to evaluate the resemblance between the cover image and the stego-image. SSIM is designed to deal with the quality of the whole image by constructing and combining equations with luminance, contrast, and structure that are recognized as the main contents in human vision. The average luminance, contrast and structural information are combined together to produce the quality index. The yield of SSIM value is limited in the range between 0 and 1. If the SSIM value is near to 1 the stego-image is alike the cover image and it has high quality. Equation (15) is used to calculate the value of SSIM:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2\mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (15)$$

where μ_x and μ_y are mean values of cover image x and stego-image y and σ_x and σ_y are standard deviation values of the cover image and stego image while, σ_{xy} means the covariance of both two images. c_1 and c_2 are constants to stabilize the division.

Like we discuss about the proposed algorithm, the blocks of the ABTC-EQ are categorized by using threshold T into smooth and complex block. Moreover, a threshold value may affect an image quality as well as an EC of DH. Table 3 shows the comparison of the performance of compression, PSNR, and SSIM in compression methods such as AMBTC, MBTC, and ABTC-Equation We know that ABTC-EQ represents better performance in all aspect of PSNR, CR, and SSIM.



Figure 4. Test images; (a–i): 512 × 512.

Table 3. Comparison of BTC-like schemes on PSNR, CR, and SSIM.

Test Images	Method	Block Size (4 × 4) Pixels			Block Size (8 × 8) Pixels		
		PSNR	CR	SSIM	PSNR	CR	SSIM
Lena	AMBTC	33.42	4.00	0.9901	29.99	6.40	0.9587
	MBTC	33.87	4.00	0.9901	30.07	6.40	0.9614
	ABTC-EQ	37.49	3.09	0.9944	34.07	4.36	0.9826
Peppers	AMBTC	33.57	4.00	0.9908	29.73	6.40	0.9619
	MBTC	34.05	4.00	0.9904	30.49	6.40	0.9629
	ABTC-EQ	37.59	3.09	0.9939	34.07	4.36	0.9818

If the threshold value increases, the image quality becomes down, while the EC of DH is increased. In other words, there exists inverse correlation between image quality and embedding capacity. In Table 4, we use the performance like EC and PSNR to compare between our proposed method and existing methods (i.e., Ou & Sun, Huang et al., and W. Hong). For a fair performance comparison, we evaluate the performance using two threshold values (i.e., 10 and 20) in every block of two compression methods, that is, AMBTC and ABTC-Equation In Table 4, we notice the performance of PSNR and EC according to the various threshold values T . There is a trade-off between EC and image quality, while it may be different in various methods.

Our proposed scheme is the first DH based on ABTC-EQ, while the existing three methods (i.e., Ou & Sun, Huang et al., and W. Hong) are based on AMBTC. Our proposed method is superior to those of existing methods in aspect of image quality. For Peppers and Tiffany images when $T = 10$,

the PSNRs of Huang et al. are higher than our proposed method by 0.2 dB. Except these two cases, the proposed method showed excellent results in two criteria namely PSNR and EC.

Table 4. Comparison of the performance using EC and PSNR of the proposed method and existing methods (where our proposed scheme is based on Adaptive BTC based on Edge Quantization (ABTC-EQ) and the others are based on AMBTC).

Images	T	Ou and Sun [19]		Huang et al. [20]		W. Hong [21]		The Proposed	
		EC	dB	EC	dB	EC	dB	EC	dB
Bridge	10	56,417	28.3860	102,905	28.3181	76,498	28.3341	202,289	31.9515
Elaine		139,425	32.0391	180,749	32.3638	163,067	31.9954	245,633	33.6242
Boats		160,913	31.0204	201,032	30.9255	186,330	30.9774	223,668	33.0478
Goldhill		127,409	31.6842	169,613	31.8073	150,349	31.6372	193,966	32.7781
Airplane		194,897	31.3173	232,886	31.2578	221,824	31.2796	256,581	33.8083
Lena		193,249	32.3724	231,347	32.4457	220,205	32.3277	254,182	34.1004
Peppers		200,369	32.2246	238,022	32.5160	227,287	32.1842	261,677	31.7616
Tiffany		197,761	33.6220	235,493	33.7989	225,488	33.5795	260,227	29.9104
Zelda		212,753	33.6013	249,632	34.0245	240,483	33.5452	270,248	34.5302
Average		164,799	31.8075	204,631	31.9397	190,170	31.7623	240,941	32.8347
Images	T	Ou and Sun [19]		Huang et al. [20]		W. Hong [21]		The Proposed	
		EC	dB	EC	dB	EC	dB	EC	dB
Bridge	20	125,137	27.0885	183,660	27.1761	148,953	27.0828	269,701	31.2579
Elaine		249,601	28.9561	300,377	29.4604	280,801	28.9606	272,186	33.1541
Boats		205,809	29.5664	259,506	29.3043	233,709	29.5557	266,650	30.3882
Goldhill		212,193	29.1224	265,479	29.4090	240,231	29.1121	268,759	29.6549
Airplane		226,977	30.1906	279,343	29.6937	256,110	30.1792	282,289	31.828
Lena		233,697	30.7508	285,651	30.3440	263,366	30.7356	290,668	31.8599
Peppers		240,977	30.6910	292,476	30.4371	271,132	30.6750	296,883	29.8852
Tiffany		241,921	31.4269	293,249	31.0589	272,691	31.4116	295,394	28.5693
Zelda		253,841	31.5579	304,536	31.2650	284,866	31.5299	303,268	32.6241
Average		221,128	29.9278	273,809	29.7943	250,207	29.9158	282,866	31.0246

Huang et al.'s method offers high EC with keeping low distortion when $T = 10$ and 20 , relatively. And, this method may be better than the previous schemes (i.e., Ou & Sun and W. Hong) because it has a code to improve image quality. But the proposed method based on ABTC-EQ can offer higher EC and PSNR than those based on AMBTC generally. The weakness of our method is that it provides somewhat higher performance for smooth images, but at some specific points PSNR falls but does not get much worse.

In Figure 5, we show the performance comparisons between the proposed method based on ABTC-EQ and the existing methods based on AMBTC, respectively. In the case of (a) Lena, (c) Zelda, and (d) Airplane in Figure 5, our proposed scheme offers good performance in aspects of EC and PSNR. When $T = 20$, in *Lena* image, ECs and PSNRs of our proposed method and Huang et al.'s method are 290,668 (31.8599 dB) and 285,651 (30.344 dB), respectively. Therefore, it shows that our proposed method is good objectively.

For all images, the PSNRs of our proposed scheme shows excellent performance compared to existing schemes for when secret bits are less than equal 100,000. According to increase secret bits, the gap of PSNR is gradually reduced between our proposed method and Huang et al.'s method.

In Figure 5b, the PSNR values of our proposed method are almost similar to those of Huang et al.'s method. After then, it is shown that the PSNR of our proposed method is reduced rather than that of Huang et al.'s method, but the gap between them is not high. In our proposed scheme, it has an ability to embed 320,000 bits maximally into cover image although it was difference by the feature of the image, while Huang et al.'s method has a capability to embed more than 340,000 bits.

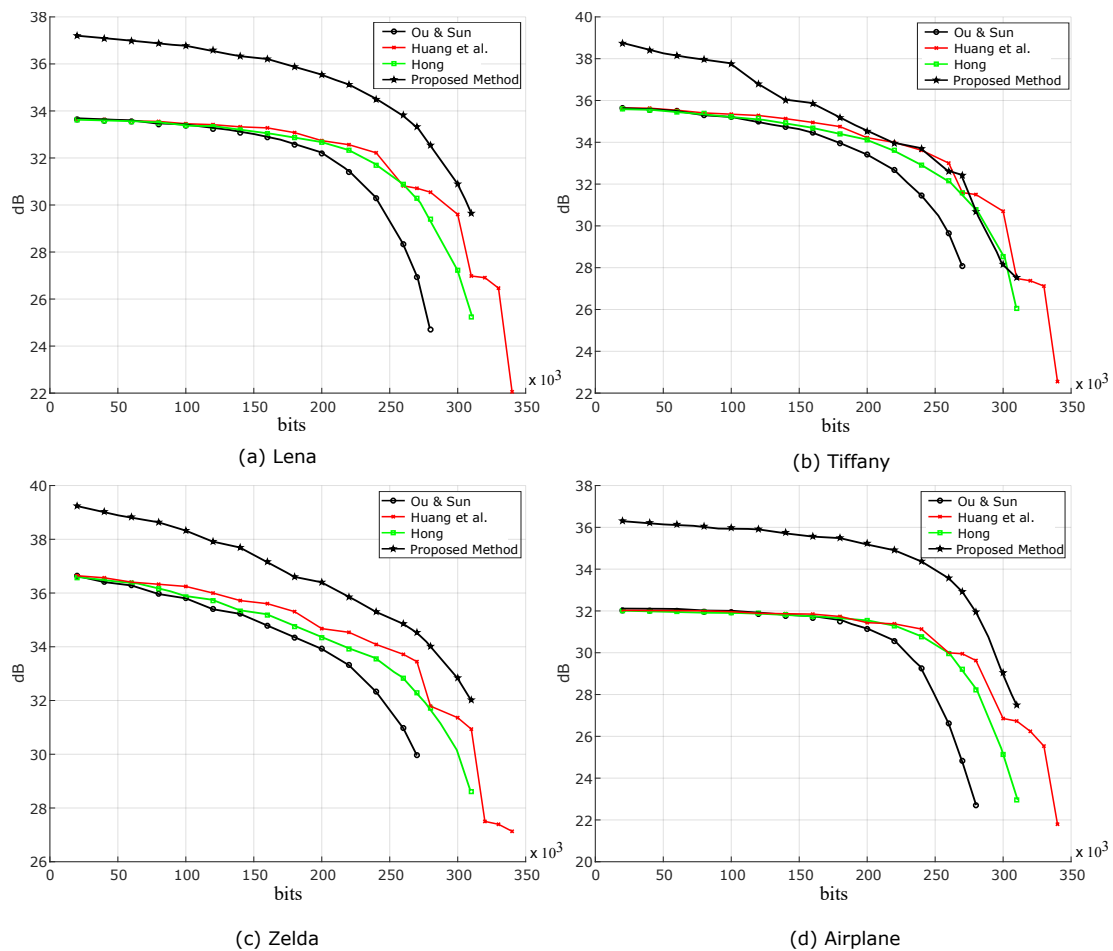


Figure 5. Performance comparisons of the proposed method and other related methods (i.e., Ou and Sun [19], Huang et al. [20], and Hong [21]) based on AMBTC and ABTC-EQ compressed image for four typical images: (a) Lena, (b) Tiffany, (c) Zelda, and (d) Airplane.

But, if more than 310,000 bits (information) will be embedded into most cover images, they become useless as cover media, since the PSNR of them become low, less than 30 dB. Huang et al.’s method is also in that same performance range. Our proposed scheme is possible to embed more than 340,000 bits if we make sacrifice PSNRs even larger using LSB3 and LSB4. In order to keep allowable PSNR, it is proved that our proposed optimization for LSB pixels is an efficient solution.

Figure 6 is shown an evaluation that we embed same number of bits (i.e., 280,000) into Lena cover image using four DH methods including our proposed method and existing methods, then compare marked Lena images using the human visual system and PSNRs. Both of (a) and (c) in Figure 6 exhibit low image quality of less than 30 dB as a result of experiment, while (b) and (d) show high image quality as 30.5441 dB and 32.5431 dB, respectively. Especially, we find out that our proposed method has highest image quality among them. Therefore, we proved that our proposed scheme is appropriate for covert communication through the simulation.



Figure 6. Visual comparison after embedding the same 280,000 bits into the Lena image by using four different methods such as (a) Ou and Sun [19], (b) Huang et al. [20], (c) W. Hong [21], and our proposed method.

5. Conclusions

In this paper, we proposed a DH method based on ABTC-EQ like BTC and used OPAP method to optimize pixels. ABTC-EQ offers high PSNR compared to AMBTC, so ABTC-EQ may be very useful cover media. In addition, the OPAP is used to reduce the loss of PSNR from the noises generated in DH process. In order to improve the efficiency for DH, each block in ABTC-EQ is categorized into the smooth and complex block and our proposed method may be applied to both types of the blocks. That is, if the block is smooth block, DBS method is applied to it and we may exchange the order of two quantization levels to hide additional one bit. The OPAP is applied to two quantization levels regardless of the categorization of the blocks. Thus, that is the explanation of the reason why our proposed scheme has high embedding capacity compared to Huang et al.'s scheme. The experimental results show that our proposed scheme offers high EC without loss of quality of a cover image.

Author Contributions: Conceptualization, C.-N.Y. and C.K.; methodology, C.K.; validation, C.-N.Y. and C.K.; formal analysis, C.-N.Y.; writing—original draft preparation, C.K. and L.L.; funding acquisition, C.-N.Y., C.K., and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China (61866028), Key Program Project of Research and Development (Jiangxi Provincial Department of Science and Technology)(20171ACE50024), Foundation of China Scholarship Council (CSC201908360075), Open Foundation of Key Laboratory of Jiangxi

Province for Image Processing and Pattern Recognition (ET201680245, TX201604002). This research was supported in part by the Ministry of Science and Technology (MOST), under Grant MOST 108-2221-E-259-009-MY2. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by (2015R1D1A1A01059253), and was supported under the framework of international cooperation program managed by NRF (2016K2A9A2A05005255).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dragoi, I.C.; Coltuc, D. Local-prediction-based difference expansion reversible watermarking. *IEEE Trans. Image Process.* **2014**, *23*, 1779–1790. [[CrossRef](#)] [[PubMed](#)]
2. Bender, W.; Gruhl, D. Techniques for data hiding. *IBM Syst. J.* **1996**, *35*, 313–336. [[CrossRef](#)]
3. Huang, F.; Qu, X.; Kim, H.J.; Huang, J. Reversible data hiding in JPEG images. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1610–1621. [[CrossRef](#)]
4. Kim, C.; Shin, D.; Yang, C.N.; Chen, Y.C.; Wu, S.Y. Data hiding using sequential hamming +k with m overlapped pixels. *KSII Trans. Internet Inf.* **2019**, *13*, 6159–6174.
5. Mielikainen, J. LSB matching revisited. *IEEE Signal Proc. Lett.* **2006**, *13*, 285–287. [[CrossRef](#)]
6. Chan, C.K.; Cheng, L.M. Hiding data in images by simple LSB substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [[CrossRef](#)]
7. Leng, L.; Li, M.; Kim, C.; Bi, X. Dual-source discrimination power analysis for multi-instance contactless palmprint recognition. *Multimed. Tools Appl.* **2017**, *76*, 333–354. [[CrossRef](#)]
8. Leng, L.; Zhang, J.S.; Khan, M.K.; Chen, X.; Alghathbar, K. Dynamic weighted discrimination power analysis: A novel approach for face and palmprint recognition in DCT domain. *Int. J. Phys. Sci.* **2010**, *5*, 2543–2554.
9. Shi, Y.Q.; Li, X.; Zhang, X.; Wu, H.T.; Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access* **2016**, *4*, 3210–3237. [[CrossRef](#)]
10. Rahmani, P.; Dastghaibifard, G. Two reversible data hiding schemes for VQ-compressed images based on index coding. *IET Image Process.* **2018**, *12*, 1195–1203. [[CrossRef](#)]
11. Delp, E.; Mitchell, O. Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342. [[CrossRef](#)]
12. Lema, M.D.; Mitchell, O.R. Absolute moment block truncation coding and its application to color images. *IEEE Trans. Commun.* **1984**, *COM-32*, 1148–1157. [[CrossRef](#)]
13. Kumar, R.; Singh, S.; Jung, K.H. Human visual system based enhanced AMBTC for color image compression using interpolation. In Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 7–8 March 2019; pp. 903–907.
14. Hong, W.; Chen, T.S.; Shiu, C.W. Lossless steganography for AMBTC compressed images. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008; pp. 13–17.
15. Mathews, J.; Nair, M.S.; Jo, L. Modified BTC algorithm for gray scale images using max-min quantizer. In Proceedings of the International Multi Conference on Automation, Computing, Control, Communication and Compressed Sensing—iMac4s 2013 (India), Kottayam, India, 22–23 March 2013; pp. 377–382.
16. Mathews, J.; Nair, M.S. Adaptive block truncation coding technique using edge-based quantization approach. *Comput. Electr. Eng.* **2015**, *43*, 169–179. [[CrossRef](#)]
17. Chuang, J.C.; Chang, C.C. Using a simple and fast image compression algorithm to hide secret information. *Int. J. Comput. Appl.* **2006**, *28*, 329–333.
18. Chen, J.; Hong, W.; Chen, T.S.; Shiu, C.W. Steganography for BTC compressed images using no distortion technique. *Imaging Sci. J.* **2010**, *58*, 177–185. [[CrossRef](#)]
19. Ou, D.; Sun, W. High payload image steganography with minimum distortion based on absolute moment block truncation coding. *Multimed. Tools Appl.* **2015**, *74*, 9117–9139. [[CrossRef](#)]
20. Huang, Y.H.; Chang, C.C.; Chen, Y.H. Hybrid secret hiding schemes based on absolute moment block truncation coding. *Multimed. Tools Appl.* **2017**, *76*, 6159–6174. [[CrossRef](#)]
21. Hong, W. Efficient data hiding based on block truncation coding using pixel pair matching technique. *Symmetry* **2018**, *10*, 36. [[CrossRef](#)]
22. Hong, W.; Chen, T.S. A novel data embedding method using adaptive pixel pair matching. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 176–184. [[CrossRef](#)]

23. Chen, Y.Y.; Chi, K.Y. Cloud image watermarking: High quality data hiding, and blind decoding scheme based on block truncation coding. *Multimed. Syst.* **2019**, *25*, 551–563. [[CrossRef](#)]
24. Malik, A.; Sikka, G.; Verma, H.K. An AMBTC compression-based data hiding scheme using pixel value adjusting strategy. *Multidimens. Syst. Signal Process.* **2018**, *29*, 1801–1818. [[CrossRef](#)]
25. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal.* **1986**, *PAMI-8*, 679–698. [[CrossRef](#)]
26. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal.* **2002**, *24*, 881–892. [[CrossRef](#)]
27. Wang, R.Z.; Lin, C.F.; Lin, J.C. Hiding data in images by optimal moderately significant-bit replacement. *IEE Electron. Lett.* **2000**, *36*, 2069–2070. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).